

A Multi-Criteria Model for Robust Foreground Extraction

A. Homayoun Kamkar-Parsi, Robert Laganière, Martin Bouchard

University of Ottawa

Ottawa, Ontario, CANADA K1N 6N5

Akamkar|Laganiere|Bouchard@site.uottawa.ca

ABSTRACT

Numerous methods are currently available for motion detection using background modeling and subtraction. However, there are still many challenges to take into account such as moving shadows, illumination changes, moving background, relocation of background objects, and initialization with moving objects. This paper provides a new background subtraction algorithm that aggregates the classification results of several foreground extraction techniques based on UV color deviations, probabilistic gradient information and vector deviations, in order to produce a single decision that is more robust to those challenges.

Categories and Subject Descriptors

I.4 [Image processing and computer vision]: Segmentation - pixel classification.

General Terms: Algorithms, Measurement.

Keywords: Visual surveillance, background subtraction.

1. INTRODUCTION

In video surveillance systems, moving object detection is an essential factor to monitor activities in indoor or outdoor environments. Detection is often achieved by using background subtraction methods. However background subtraction is just the first stage in a video surveillance system. The results obtained from this stage are used for further processing, such as target tracking. Background subtraction methods build a model of the background scene from the video signal of a fixed camera. For each pixel in the received image, the methods detect deviations of pixel values from the model to classify the pixel as belonging either to the background or the foreground scene. The foreground scene consists of transient objects passing through the scene or motion involving the observed objects. The proposed background subtraction algorithm combines different detection techniques to proceed to foreground/background classification.

2. ALGORITHM

The entire algorithm is based on the YUV color space. It allows the separation of the color information (i.e. chroma channel blue, $I_U^t(i, j)$, and the chroma channel red, $I_V^t(i, j)$) from the luminance component, $I_Y^t(i, j)$, for a given pixel $I^t(i, j)$ at time instant t . The background is modeled statistically on a pixel-by-

pixel basis. Each background pixel is modeled by 4 statistical parameters: the mean luminance value, $B_Y^t(i, j)$, the variance of the luminance value, $\sigma_Y^{2,t}(i, j)$, and the mean of the blue and red chroma channels, $B_U^t(i, j)$, $B_V^t(i, j)$. The luminance component of each background pixel is modeled by a Gaussian distribution. Indeed, in a completely static scene under fixed lighting, the pixel intensity over time can be rationally modeled with a Gaussian distribution $N(\mu, \sigma^2)$, given the acquisition of the image noise (for example: camera noise) over time which could typically be modeled by a zero mean Gaussian distribution $N(0, \sigma^2)$. However, the Gaussian distribution model has not been applied to the color components since their variances are negligible compared to the variance of the luminance component. Under illumination variations, the color components are less affected in comparison to the luminance component; only the means of the color values, $B_U^t(i, j)$ and $B_V^t(i, j)$, are used.

2.1 Detection based on UV color deviations

For the first detection technique, the Euclidean distance between each incoming pixel color component and the corresponding background pixel mean component, is calculated to classify each pixel as a foreground, FG , or a background pixel, BG , i.e.:

$$\text{if } \left(I_U^t(i, j) - B_U^t(i, j) \right)^2 + \left(I_V^t(i, j) - B_V^t(i, j) \right)^2 > T_C, \\ \text{then } I^t(i, j) \Rightarrow FG_1 \text{ else } I^t(i, j) \Rightarrow BG_1 \quad (1)$$

The threshold T_C represents the magnitude of the color deviation from the background model. The subscripts for FG_1 and BG_1 indicate that $I^t(i, j)$ has been classified by the selected technique (i.e. technique #1 for this case). Before applying this technique, the saturation value for each incoming pixel and corresponding background pixel should be evaluated. If both saturation values are low, the detection technique using (1) is not applicable because the color information obtained from the U and V channels is irrelevant. The saturation value, SV , for a given background pixel could be evaluated as follows:

$$SV = \sqrt{\left(B_U^t(i, j) - W_U \right)^2 + \left(B_V^t(i, j) - W_V \right)^2} \quad (2)$$

where W_U and W_V correspond to the U and V color channel values of reference white. The same equation is used to evaluate the saturation value for the incoming pixel by replacing $B_U^t(i, j)$ with $I_U^t(i, j)$ and $B_V^t(i, j)$ with $I_V^t(i, j)$, respectively. If both saturation values are below a selected threshold, T_S , a valid decision cannot be made, and the current pixel, $I^t(i, j)$, is then classified as undetermined (DKN_1).

2.2 Detection based on gradient deviations

The second detection technique uses the gradient information. It creates a statistical model of gradient distributions for each pixel

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

VSSN'05, November 11, 2005, Singapore.

Copyright 2005 ACM 1-59593-242-9/05/0011...\$5.00.

of the background model. A Gradient-based subtraction approach has also been proposed in [1]. Here, we propose to model the probability of appearance and disappearance of edges due to moving objects in the scene. Our probability model is similar to the Order Consistency criteria described in [2].

For each pixel, the horizontal and vertical gradient background distributions are computed using luminance values of the background model. Consequently, each gradient will also follow Gaussian distribution with mean $G_H^t(i, j)$ and variance $\sigma_H^{2,t}(i, j)$ for the horizontal gradient distribution, and with $G_V^t(i, j)$ and $\sigma_V^{2,t}(i, j)$ for the vertical gradient distribution. These values are computed as follows:

$$\begin{aligned} G_H^t(i, j) &= B_Y^t(i, j+1) - B_Y^t(i, j-1) \\ \sigma_H^{2,t}(i, j) &= \sigma_Y^{2,t}(i, j+1) + \sigma_Y^{2,t}(i, j-1) \end{aligned} \quad (3)$$

$$\begin{aligned} G_V^t(i, j) &= B_Y^t(i+1, j) - B_Y^t(i-1, j) \\ \sigma_V^{2,t}(i, j) &= \sigma_Y^{2,t}(i+1, j) + \sigma_Y^{2,t}(i-1, j) \end{aligned} \quad (4)$$

From these models, the probability of observing vertical ($P_H^t(i, j)$) and horizontal ($P_V^t(i, j)$) edges can be computed. For instance, $P_H^t(i, j)$ is computed as:

$$\begin{aligned} P_H^t(i, j) &= \Pr(x < -T_G) + \Pr(x > T_G) \\ &= \Pr\left(\frac{x - G_H^t(i, j)}{\sqrt{\sigma_H^{2,t}(i, j)}} < \frac{-T_G - G_H^t(i, j)}{\sqrt{\sigma_H^{2,t}(i, j)}}\right) + \\ &\quad \left(1 - \Pr\left(\frac{x - G_H^t(i, j)}{\sqrt{\sigma_H^{2,t}(i, j)}} < \frac{T_G - G_H^t(i, j)}{\sqrt{\sigma_H^{2,t}(i, j)}}\right)\right) \end{aligned} \quad (5)$$

where T_G is a selected gradient threshold representing the minimum magnitude difference of luminances from which two pixels located at $(i+1, j)$ and $(i-1, j)$ or $(i, j+1)$ and $(i, j-1)$ could be heuristically considered as an edge pixel at (i, j) . $P_V^t(i, j)$ is also computed using (5) but using the vertical mean and variance instead.

For each incoming frame, the instantaneous horizontal and vertical gradients $I_{G,H}^t$ and $I_{G,V}^t$ for each pixel $I^t(i, j)$ are computed as:

$$\begin{aligned} I_{G,H}^t &= I_Y^t(i, j+1) - I_Y^t(i, j-1) \\ I_{G,V}^t &= I_Y^t(i+1, j) - I_Y^t(i-1, j) \end{aligned} \quad (6)$$

The current new pixel can then be considered as new vertical edge pixel (i.e. appearance of an edge) not present in the background model if $|I_{G,H}^t|$ is higher than T_G and its corresponding horizontal background gradient $|G_H^t(i, j)|$ is lower than T_G . Reciprocally, the current pixel could potentially be covering a previous background edge pixel (i.e. disappearance of an edge) if $|I_{G,H}^t|$ is lower than T_G and its corresponding horizontal background gradient

$|G_H^t(i, j)|$ is higher than T_G . The explanation above is summarized by the following expression:

$$\text{if } |G_H^t(i, j)| < T_G \ \& \ |I_{G,H}^t(i, j)| > T_G \ \text{then} \quad (7) \\ \Rightarrow App = 1, Disp = 0$$

$$\text{else if } |G_H^t(i, j)| > T_G \ \& \ |I_{G,H}^t(i, j)| < T_G \ \text{then} \\ \Rightarrow Disp = 1, App = 0$$

where $App=1$ represents the potential ‘‘appearance’’ of a vertical edge pixel, and $Disp=1$ represents the potential ‘‘disappearance’’ of a background vertical edge. The same procedure is performed for the vertical direction but using $I_{G,V}^t$ and $G_V^t(i, j)$ instead.

From the probability results found using (5), the probability of either an appearance or a disappearance of a vertical edge pixel, $P_{H, App \parallel Disp}$, is evaluated as follows:

$$\begin{aligned} P_{H, App \parallel Disp} &= \Pr(App = 1 \ \text{or} \ Disp = 1) \\ &= \min\left(P_H^t(i, j), 1 - P_H^t(i, j)\right) \end{aligned} \quad (8) \quad \text{By}$$

replacing $P_H^t(i, j)$ by $P_V^t(i, j)$ in (8), the probability of appearance or disappearance of an horizontal edge, $P_{V, App \parallel Disp}$, is also evaluated. Finally, the classification procedure is performed as in (9). Starting for the horizontal gradient analysis using equations (6-8) we have:

$$\begin{aligned} \text{if } (App = 0 \ \& \ Disp = 0) \ \text{then} \quad I^t(i, j) &\Rightarrow BG_2 \\ \text{else if } (App = 1 \ \text{or} \ Disp = 1) \ \& \ (P_{H, App \parallel Disp} < T_{App \parallel Disp}) \\ \text{then } I^t(i, j) &\Rightarrow FG_2 \ \text{else } I^t(i, j) \Rightarrow DKN_2 \end{aligned} \quad (9)$$

The threshold probability, $T_{App \parallel Disp}$, should be much lower than 50% (i.e. in the range of 10-20%). For instance if $P_{H, App \parallel Disp}$ resulting from (8) is equal to 50% at location (i, j) , it implies that the mean gradient, $G_H^t(i, j)$, from the background gradient model distribution, is equal to $|T_G|$. Consequently, a small perturbation on the incoming pixels would toggle the labels App and $Disp$. Therefore, there is not enough evidence of either a real appearance or a real disappearance of a pixel, and $I^t(i, j)$ should then be classified as an undetermined (DKN_2) pixel instead of a foreground pixel. Then, if for the horizontal analysis, the classification for $I^t(i, j)$ yields either a background pixel (BG_2) or an undermined pixel (DKN_2), then the same analysis should also be done with the vertical gradient, using again equations (6-8).

2.3 Detection based on vector deviations

The third detection technique creates a vector from 5 neighbour pixels, $V_I^t(i, j)$, for each incoming pixel in the frame and compares it to the corresponding vector, $V_B^t(i, j)$, formed using the background model:

$$V_I^t(i, j) = \left[I_Y^t(i, j), I_Y^t(i+1, j), I_Y^t(i-1, j), I_Y^t(i, j+1), I_Y^t(i, j-1) \right] \quad (10)$$

$$V_B^t(i, j) = \left[B_Y^t(i, j), B_Y^t(i+1, j), B_Y^t(i-1, j), B_Y^t(i, j+1), B_Y^t(i, j-1) \right] \quad (11)$$

This kind of vector model is also used in the Grammian change detector [3]. Here, the comparison is achieved by calculating the angle between the vectors as shown in (13) and by comparing their length as in expression (14) defined in the next section. The angular deviation criterion is defined as follows:

$$\theta = \cos^{-1} \left(\frac{V_I^t(i, j) \bullet V_B^t(i, j)}{\|V_I^t(i, j)\| \|V_B^t(i, j)\|} \right) \quad (12)$$

$$\text{if } \theta > T_{\text{angle}} \text{ then } I^t(i, j) \Rightarrow FG_3 \quad (13)$$

$$\text{else } I^t(i, j) \Rightarrow BG_3$$

where \bullet corresponds to the dot product of 2 vectors and $\| \cdot \|$ is the magnitude of a vector. T_{angle} is the threshold corresponding to the magnitude of the angle deviation. It should be noted that with this technique, three distinct scenarios would yield a pixel classification as a background:

i) if the scene did not change; the input vector received is then a background vector;

ii) if there is a change of illumination; for instance a shadow occurs in the scene, the vectors would still have similar orientation but the input vector would have a smaller magnitude;

iii) in the case of a uniform background covered by an object with a different but uniform color as well. This foreground object would be classified as background since the vectors would be pointing in the same direction. This scenario would yield a false classification. However, it can be corrected with the following and last technique that consists in comparing the vector magnitudes.

2.4 Detection based on vector magnitude ratio

The fourth technique is a complement of the third technique since it only processes the background-classified pixels (BG_3) outputs from the third technique. It calculates the ratio of the magnitudes of the vectors that have similar orientations (i.e. $I^t(i, j) \Rightarrow BG_3$). This technique can also be used for the undetermined-classified pixels (DKN_1) obtained from the first technique in section 2.1 (low saturation case).

When the magnitude ratio of these vectors fall within $[T_{LB}, T_{UB}]$, then their difference is assumed to be due to cast shadows (causing local illumination changes on the background) or to other form of illumination fluctuations (e.g. sun/cloud transitions) and consequently the current pixel would be then correctly classified as BG_4 , that is:

$$\text{if } (I^t(i, j) \Rightarrow DKN_1 \text{ or } I^t(i, j) \Rightarrow BG_3) \text{ then}$$

$$\text{ratio} = \frac{\|V_I^t(i, j)\|}{\|V_B^t(i, j)\|}$$

$$\text{if } (T_{LB} < \text{ratio} < T_{UB}) \text{ then } I^t(i, j) \Rightarrow BG_4$$

$$\text{else } I^t(i, j) \Rightarrow FG_4 \quad (14)$$

where T_{LB} is the lower bound threshold and T_{UB} is the upper bound threshold of the ratio of the vector's magnitudes.

2.5 Aggregation of the techniques

Each technique presented here produces independent foreground segmentation results. This section will show how to combine all the results in order to produce a single decision corresponding to the classification of the incoming pixel as belonging either to foreground or background. Figure 1 shows the block diagram on how to combine the techniques. The symbol \oplus represents the binary logic "or". Each arrow indicates an output that could be either FG or BG represented by binary logics "1" and "0", respectively. Only technique #2 yields 5 types of outputs: BG , FG , DKN , App and $Disp$. In general, at stage A , the most foreground-classified pixels correspond to the "body" of moving objects (i.e. dense mapping) and at stage B , most foreground-classified pixels correspond to the edges of moving objects. The

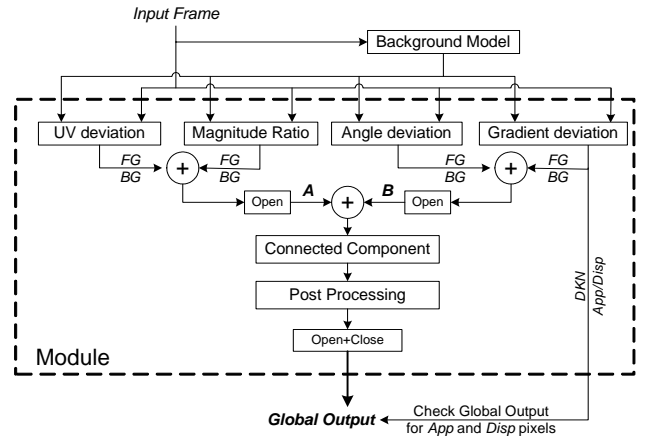


Figure 1: Block diagram representing the aggregation of the techniques and the post processing

output of the connected component block (8-connectivity has been used) groups the foreground-detected pixels into regions, referred to as blobs (i.e. moving objects). After finding all the blobs, the post processing block analyses separately each blob and decides whether or not to keep the blob. Its decision making process is explained in (15). It can be seen that the entire region (i.e. blob) is discarded if its total area, $Area(blob)$ is lower than a minimum selected number of pixels, $T_{MinArea}$. The entire blob is also discarded if the number of foreground-classified pixels from A belonging to the current blob, $Area(A \in Blob)$, is lower than a minimum percentage, T_{AreaL} , of $Area(blob)$. Also, it has been observed that motion background (such as tree leaves under the wind) tends to produce lot of FG pixels of category A (i.e. perturbed edges). In this case, the number of foreground classified pixels from B belonging to the blob but different than the ones from A , $Area((B \notin A) \in Blob)$, will be higher than a maximum percentage T_{AreaH} of $Area(blob)$. The solution is therefore to keep only a portion of the entire blob (i.e. the foreground classified pixels from A belonging to the blob, $A \in Blob$). Finally, if $Area(A \in Blob)$ is higher than the minimum percentage, T_{AreaL} , of $Area(Blob)$, then the entire blob is kept.

if $Areal(Blob) > T_{MinArea}$ {
 if $Area(A \in Blob) < T_{AreaL} \cdot Area(Blob)$,
 \Rightarrow discard entire blob
 else if $Area((B \notin A) \in Blob) > T_{AreaH} \cdot Area(Blob)$,
 \Rightarrow keep only $A \in Blob$
 else if $Area(A \in Blob) > T_{AreaL} \cdot Area(Blob)$,
 \Rightarrow keep entire Blob
 }

} else \Rightarrow discard entire blob (15)

Global Output represents the overall classification assigned to the incoming pixel (FG or BG). Also, as illustrated in Fig.1, its foreground-classified pixels can be checked with the output of technique#2, to further differentiate between an appearance of an edge pixel and the disappearance of a background edge pixel.

3. MODEL UPDATING

Following the idea in [4], the background model is being updated using an exponential averaging, where α is the learning rate.

$$B_Y^{t+1}(i, j) = (1 - \alpha) \cdot B_Y^t(i, j) + \alpha \cdot I_Y^t(i, j), \quad (16)$$

$$B_U^{t+1}(i, j) = (1 - \alpha) \cdot B_U^t(i, j) + \alpha \cdot I_U^t(i, j),$$

$$B_V^{t+1}(i, j) = (1 - \alpha) \cdot B_V^t(i, j) + \alpha \cdot I_V^t(i, j),$$

$$\sigma_Y^{2,t+1}(i, j) = (1 - \alpha) \cdot \sigma_Y^{2,t}(i, j) + \alpha \cdot (I_Y^t(i, j) - B_Y^t(i, j))^2$$

In order to resolve the problems of bootstrapping, relocation of background objects, foreground objects becoming stationary and sudden illumination changes, the algorithm creates a secondary background model (BM#2). It also creates and updates a

foreground model (FM) with its 4 statistical parameters: $F_Y^t(i, j)$,

$\sigma_F^{2,t}(i, j)$, $F_V^t(i, j)$ and $F_U^t(i, j)$; these ones being updated as in

(16). The second background model (BM#2) is modeled like the first background model (BM#1) with again 4 statistical parameters. For each incoming frame, the different segmentation techniques illustrated in the block diagram of Fig. 1 will also be performed using instead BM#2 as a reference. Figure 2 illustrates the overall algorithm structure. Each module output (i.e. *Global output # 1* or # 2) results from the aggregation of the techniques using their corresponding background models, BM#1 and BM#2. However, for each incoming frame, only one of the background models will be selected for an update (including the selection criteria) as described in (17). For each incoming

frame, $BG_{cnt}^1(i, j)$ and $BG_{cnt}^2(i, j)$ count the number of frames that a given pixel at (i, j) has been classified as BG by module #1 and #2, respectively. The foreground model parameters are updated only when a pixel at (i, j) is FG -classified by both *Global Outputs*. $FG_{cnt}(i, j)$ counts the number of frames that the pixel at (i, j) remained foreground-classified. If it is classified as a foreground over a selected number of frames or seconds, T_{wait} , it should become part of the background model by replacing the background model having the lowest background-classified counter value (i.e. $BG_{cnt}^1(i, j)$ or $BG_{cnt}^2(i, j)$) with the current foreground model. The FM parameters are also reset.

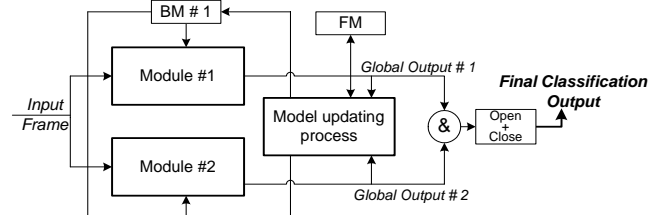


Figure 2: Overall algorithm structure

if $Global\ Output\ #1 \Rightarrow BG$ then,

$$FG_{cnt}(i, j) = 0, BG_{cnt}^1(i, j) = BG_{cnt}^1(i, j) + 1;$$

Update BM#1 parameters using (16);

else if $Global\ Output\ #2 \Rightarrow BG$ then,

$$FG_{cnt}(i, j) = 0, BG_{cnt}^2(i, j) = BG_{cnt}^2(i, j) + 1; \quad (17)$$

Update BM#2 parameters using (16);

else

$$FG_{cnt}(i, j) = FG_{cnt}(i, j) + 1;$$

Update FM parameters using (16);

if $FG_{cnt}(i, j) = T_{wait}$ then,

Replace BM having min. BG_{cnt} with FM Parameters;

$FG_{cnt}(i, j) = 0$, reset FM Parameters;

$$BG_{cnt}^1(i, j) = BG_{cnt}^2(i, j) = 0;$$

4. CONCLUSION

An efficient and effective algorithm for foreground extraction has been described. It combines different complementary techniques to perform robust background subtraction of the visual data collected by a fixed camera. Chrominance information is first used because it allows reliable positive detection of foreground parts. More foreground elements are then detected by considering neighboring pixel information, integrated in a vector form and compared to the background model in an illumination-invariant way. For more accurate foreground extraction, the algorithm also incorporates a probabilistic formulation for gradient-based change detection. All these methods are combined in a way that makes foreground extraction invariant to illumination changes (e.g. shadows), object relocation and small background motion.

5. REFERENCES

- [1] Javed, O., Shafique, K., and Shah M. A Hierarchical Approach to Robust Background Subtraction using Color and Gradient Information. *In Proceedings of the Workshop on Motion and Video Computing*, 2002, 22-28.
- [2] Xie, B., Ramesh, V., and Boulton, T. Sudden Illumination change detection using Order Consistency. *Image and Vision Computing*, 22, 2, (Feb. 2004), 117-125.
- [3] Durucan, E., and Ebrahimi, T. Change Detection by Nonlinear Grammian, in *Proceedings of NSIP*, 2001.
- [4] Stauffer, C., and Grimson, W. Learning Patterns of Activity Using Real-Time Tracking, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22, 8, (2000), 747-757